

⑫ 公開特許公報(A)

昭61-84735

⑤Int.Cl.⁴
G 06 F 7/00識別記号
1 0 2庁内整理番号
7313-5B

④公開 昭和61年(1986)4月30日

審査請求 未請求 発明の数 1 (全1頁)

⑬発明の名称 汎用レジスタの有効長拡張装置

⑭特 願 昭59-206235

⑮出 願 昭59(1984)10月3日

⑯発明者 後藤 志津雄 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑰発明者 鍵政 豊彦 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑱発明者 吉住 誠一 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑲発明者 新谷 洋一 国分寺市東恋ヶ窪1丁目280番地 株式会社日立製作所中央研究所内
⑳出願人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地
㉑代理人 弁理士 高橋 明夫 外1名

明 細 書

発明の名称 汎用レジスタの有効長拡張装置
特許請求の範囲

汎用レジスタを基本とする命令体系を有する情報処理装置において、夫々の汎用レジスタの有効長を指示する有効長指示モード記憶手段と、汎用レジスタを扱う命令の動作時に汎用レジスタから読み出す情報の長さ、汎用レジスタに書き込む情報の長さ、あるいは汎用レジスタを用いる演算の対象とする情報の長さを、上記のモード記憶手段に記憶された値に従って変更する手段とを有することを特徴とする汎用レジスタの有効長拡張装置。

発明の詳細な説明

〔発明の利用分野〕

本発明は汎用レジスタを基本とする情報処理装置に係り、特に汎用レジスタの従来の長さを超えるようなアドレスやデータを取り扱うことができるような汎用レジスタの有効長の拡張を、命令体系の大巾な変更を伴わずに実現するのに好適な汎用レジスタの有効長拡張装置に関する。

〔発明の背景〕

情報処理装置の処理するデータ量の増大に伴って、情報の処理能力を増大させようという要請が生じる。その1つとしてアドレスの上限値の拡張、すなわち、プログラムからアクセスできるデータの量を増大(アドレス拡張と呼ぶ)させようという要請がある。他の1つは、演算処理単位の増大(データ拡張と呼ぶ)であり、1命令で処理する情報の処理単位を増加させようという要請である。これらの要請を満たす情報処理装置の作り方として、次の2つがある。1つは新しい命令体系を持つ情報処理装置を作る方法であり、他の1つは従来の情報処理装置に若干の機能追加をする方法である。ところで1つの命令体系の下で数多くのプログラムが既に作成されている場合、それらが新しい情報処理装置においてもそのまま動作しうることも要請される。この点を考えると上記の中では後者の方法、すなわち従来の命令体系に若干の修正をし、従来のプログラムを変更することなく動作させることができ、かつ新しいプログラムも

動作させることができる情報処理装置を作る方が良策である。

従来プログラムをそのまま動作させることができ、かつアドレス拡張を実現した方式としてはIBM社の「IBM System/370 Extended Architecture Principles of Operations」(SA-22-7085-0)に示されているような方式がある。そこではプログラムの実行を制御するためのプログラム状態語(PSW)の中にアドレス拡張ビットを設け、その値が0の時にはアドレスの上限値は $2^{31}-1$ 、その値が1の時にはアドレスの上限値は $2^{32}-1$ となる。以下、本方式をさらに詳細に説明する。第1図は、情報処理装置の命令形式の分類を表わしたものである。図において、 $R_1, R_2, R_3, B_1, B_2, X_1$ はすべて、16個ある32ビットの汎用レジスタの番号を指示するために用いられている。この命令体系では汎用レジスタの番号を指示するのに4ビットが使われており、16個の汎用レジスタの中のどれかを指し示すのに適した命令体系となつている。汎

用レジスタの値はアドレスであつたり、数を表現したり、あるいは文字コードであつたりするが、汎用レジスタにはそれが何を表現しているかの区別はなく、命令語の中のどの位置で指示されるかにより、その用途が決定される。すなわち、第1図において、 B_1, B_2 は、ベース・レジスタとしての使われ方であり、 B_1, B_2 の指す汎用レジスタの値と D_1, D_2 の12ビットの内容が加算された値がメモリ上のアドレスを指すのに用いられる。 RX 形式における X_1 はインデックス・レジスタとしての使われ方であり、 X_1 の指す汎用レジスタの値および B_1 の指す汎用レジスタの値および D_1 の内容の3者の加算値がメモリ上のアドレスを指すのに用いられる。 R_1, R_2, R_3 はその指す汎用レジスタの値をアドレスとして用いたり、アドレスではないデータとして用いたりするが、その用いられ方は命令語によって一意に決められている。ところで汎用レジスタの大きさは32ビットであり、従来アドレス・モードの時には下位の24ビットが有効となり上位8ビ

ットの値は無視されるような命令体系であつた。汎用レジスタへのアドレスの設定は、メモリ上の32ビットのデータを汎用レジスタにセットするLOAD命令や、32ビットの汎用レジスタのデータとメモリ上の32ビットのデータとを加算し汎用レジスタにセットするADD命令、などを用いて行つている。すなわち、32ビットのデータを基本とする命令体系を利用してアドレスの設定・計算を行つており、アドレスとして利用する時には上位8ビットを無視する操作を行つている。

このような命令体系の下でアドレスを拡張する場合を考える。1つは32ビット以内のある長さに拡張する場合であり、他の1つは32ビットを超えた長さに拡張する場合である。前述のIBM System/370 Extended Architectureではアドレスを31ビットに拡張しているが、上述のように拡張アドレスモードか否かを表わす拡張アドレス・ビットの値に従つて32ビットの汎用レジスタのうち上位8ビットを無視するが上位1ビットを無視するかを切換えることにより、従来プログラ

ムと拡張プログラムの両者を1つの情報処理装置の上で実行することが可能となる。

しかし、他の1つの拡張(例えば64ビットに拡張)を上記の方式により実現することはできない。上記の方式の拡大適用の例として、汎用レジスタの長さを64ビットに拡大し、ベース・レジスタなどアドレスとして用いられる時には64ビットの計算をする方式が考えられる。しかし、上記のような命令体系の下では演算は32ビットのため64ビットのアドレス・データの設定や計算ができない。そのため4ビットのアドレス・データの設定と計算のための命令群が新たに必要となるが、これは命令体系の大巾な変更が必要となり、好ましくない。また、拡張アドレスの設定と計算を従来の命令体系の下で実施可能とする方式として、拡張アドレス・ビットの値によつて従来の場合は32ビットの演算を行い、拡張の場合は64ビットの演算を行うようにする方式も考えられる。しかし、この方式を適用すると、従来の32ビットのデータの演算が拡張アドレス・モードの時に

はできなくなるという問題が生じる。

以上、アドレスの長さの拡張を、従来の汎用レジスタの長さを超えて実施する上での問題点について説明した。

アドレスの拡張と並び要請されるのがデータ演算の単位の拡張(データ拡張)である。汎用レジスタの長さが32ビットの従来の命令体系の下では32ビットを超える(例えば64ビット)の演算を行う命令はなかった。従来、浮動小数点レジスタについては、32ビット、64ビット、128ビットの演算をそれぞれ別々の命令として設けることにより実施可能であつたが、汎用レジスタを使う演算(固定小数点データの演算)については、連続番号のレジスタを同時にメモリからへの転送することを出してできず、64ビットの固定小数点の演算を行うには32ビットのための演算命令を多数用いて実施しているのが現状である。例えばFORTRAN などの高級言語において、処理すべきデータ量の増大に伴つて64ビットの固定小数点データを扱う要請がある(例えば、配列の大き

拡張データの演算を行い、拡張されていないデータの演算は従来と同じ長さで実施する必要がある。このためには、個々の汎用レジスタに対応して、それが拡張されたアドレス計算やデータ演算を行うか否かを表わすモード・ビットを設けるのが良い。そして、当該モード・ビットの値に従つてアドレス計算や演算の有効長を切替えることができるアドレス計算回路、演算処理回路を設け、汎用レジスタの保持している情報の長さに応じた計算や演算を実施できるようにする。

また、汎用レジスタの個数は有限であり、プログラムの実行に従つて個々の汎用レジスタの内容は変化していく。例えば最初は従来の長さのデータを保持し32ビットの演算のために用いられる汎用レジスタが、次には拡張された長さのアドレスを保持し、拡張されたメモリ範囲内のデータをアクセスするために使用され、さらに次には、拡張された長さのデータの演算を使用されるということがありうる。このような使用方法の変更をプログラム内で実行の途中で行えるように、当該拡張

さが2³²を超えるものが要求されている。)が、64ビットの演算のためにはコンパイラは多数の命令語に翻訳しなければならず処理を複雑にするとともに、処理性能を悪化させる要因となる。

以上、アドレス拡張とデータ拡張を従来の汎用レジスタの長さを超える程度に拡張する場合の問題について述べた。

(発明の目的)

本発明の目的は、命令体系の変更を殆ど行わずに、汎用レジスタの有効長を拡張し、従来のデータ長よりも大きいようなアドレスやデータを取り扱えるような、従来装置に対して変更の少ない情報処理装置を提供することにある。

(発明の概要)

汎用レジスタを基本とする命令体系の下でアドレス拡張やデータ拡張を行うには、汎用レジスタの長さを増大する必要がある。従来プログラムはそのまま実行でき、かつ新規拡張プログラムでは拡張されたアドレスやデータを扱う時のみ拡張された汎用レジスタを用いて拡張アドレスの計算や

拡張ビットの変更ができるようにする必要がある。

また、マルチタスク処理を行うオペレーティング・システムの下では、複数のプログラムの間で処理を切替えるのが普通であり、汎用レジスタは切替えの際に状態が退避回復されるので、当該拡張モード・ビットをオペレーティング・システムによる退避・回復ができるようにする必要がある。

なお、このモード・ビットは従来の命令によつて変更されるようにする必要があるので、汎用レジスタの外に別に設けるのがよい。

(発明の実施例)

以下、本発明の一実施例を第2図より第9図を用いて説明する。

第2図は、本発明が適用される情報処理装置の構成を示す。それは次のものから構成される。

(1) 命令解析・制御部 100

命令語を解説するとともに、他の部分の処理を制御する。

(2) 命令実行部 200

命令語の実行を担当する部分であり、本発明

が適用される部分である。

(3) メモリ制御部300

主記憶装置(MS)400とのデータ転送を制御する。

以上の構成は一般的なものである。

次に本発明が適用される命令実行部200内の構成を記す。

(1) 汎用レジスタ制御部10

16本の汎用レジスタ11からの読出し、およびそれらへの書き込みを制御する部分である。

(2) アドレス計算部20

メモリ・アドレスを計算する部分である。

(3) 演算処理部30

メモリから読込んだデータ、汎用レジスタから読込んだデータに対する演算を実行する部分である。

(4) モード・レジスタ制御部40

汎用レジスタの有効長を指示するモード・レジスタ41を制御する部分である。

上記のうち、汎用レジスタ制御部10、アドレ

スには命令解析・制御部100の制御の下で、命令実行部200で計算が行なわれる。すなわち、次のような処理が行われる。

- (a) 命令語内のB_i部分の内容を汎用レジスタ制御部10およびモード・レジスタ制御部に伝え、B_iの指示する汎用レジスタおよびモード・ビットをそれぞれ読み出し、さらに、アドレス計算部20内のB_iレジスタ・データ26およびB_iモード・ビット124に設定する。
- (b) (a)と同様にして、X_i部分に対応して、X_iレジスタ・データ27、およびX_iモード・ビット125に値を設定する。
- (c) 命令語内のD_i部分の内容をアドレス計算部20内のD_iデータ・レジスタ28に設定する。
- (d) B_iレジスタ・データ26、X_iレジスタ・データ27、D_iデータ・レジスタ28の3者を入力とするアドレス・アダー25により加算が行われ、結果がアドレス・レジスタ

計算部20、演算制御部30は、従来より設けられている部分であるが、モード・レジスタ制御部40は新規な部分である。これら4部分の詳細については後述することとし、ここでは各部の動作の概要を説明する。そのために、代表的な命令として、LOAD命令とADD命令の処理の流れを第3図を用いて説明する。命令の実行は通常、ステージと呼ばれる時間インターバルに分けて行われる。ここでは4つのステージに分ける場合を説明するが、本発明はこれに限定されるものではない。ステージごとに次のように処理される。

(1) 命令デコード・ステージ(D)

命令語内のオペレーション・コードを解釈し、LOAD命令、ADD命令などを認識するとともに、命令語内の各パートを取り出す。この処理は命令解析・制御部100で実施されるが、このための回路構成は公知であるので説明は省略する。

(2) アドレス計算ステージ(A)

命令語内のB_i、X_i、D_iパートを基に、第2オペランドのアドレスを計算する。具体的

に123に設定される。この際にB_iモード・ビット124、X_iモード・ビット125は、有効長の制御に使用されるが、詳細は後述する。

(3) オペランド・ロード・ステージ(L)

第2オペランドのデータをメモリからフェッチする。具体的には、メモリ・レジスタ123の内容がメモリ制御部300に転送され、MA400に対するフェッチ要求が発行される。MA400からのフェッチが行われるとフェッチ・データ・レジスタ302にデータが設定される。なお、この過程においては、仮想アドレス機構のある場合にはアドレス・レジスタ123の内容は仮想アドレスを表現しており、それを実アドレスに変換する回路が動作する。また、主記憶装置MS上のデータの写しを格納する高速のバッファ記憶装置BSを持つ場合には、BSをアクセスする回路が動作する。本実施例にはこれらについての説明は省略しているが、本発明の適用においてはこれらの回路の有無は

無関係である。

(4) 命令実行ステージ(D)

LOAD命令の場合には、フェッチしたデータをR₁ パートで指す汎用レジスタに設定する。
ADD命令の場合には、R₁ レジスタの内容とフェッチしたデータとの加算が行われ、結果をR₁ レジスタに設定する。この処理は次のように行われる。

- (a) 命令解析・制御部100で取り出されたR₁ パートの内容が汎用レジスタ制御部10およびモード・レジスタ制御部40に伝達される。
- (b) モード・レジスタ制御部40ではR₁ パートの値に従ってモード・ビットが読み出される。
- (c) 演算処理部30内のメモリ・フェッチ・データ・レジスタ31に、フェッチしたデータ302の内容が転送される。LOAD命令の場合には演算器32をバイパスして結果レジスタ137にデータが転送される。ADD命令の

場合には汎用レジスタ制御部10内でR₁ パートの指す汎用レジスタの内容が読出され、レジスタ出力データ・レジスタ131に転送され、フェッチ・データ・レジスタ31とともにALU32の入力となり、加算が行われる。加算結果は結果レジスタ137に設定される。(b)で読み出されたR₁ モード・ビット136は、結果レジスタ137の有効長制御に用いられるが、これについては後述する。

- (d) LOAD命令、ADD命令ともに、結果レジスタ137の内容が汎用レジスタ制御部10に転送され、レジスタ入力レジスタ12を介してR₁ パートの値で指定される汎用レジスタに設定される。

以上のようにして、LOAD命令、ADD命令の処理が実行される。以上の動作は、モード・ビットに関係する記述を除いて公知のものである。

次に、本発明の適用時の動作をそのための回路について詳細に説明する。まず、本実施例の動作

の詳細を、第4図を用いて、LOAD命令について説明する。

本命令1は、X₁ の指す汎用レジスタの値、B₁ の指す汎用レジスタの値、およびD₁ の12ビットの値の3者の加算値をメモリのアドレスとしてデータを読み込み、そのデータをR₁ で指す汎用レジスタにセットする命令である。従来モードにおいては、アドレスおよびデータの長さは4バイト(=32ビット)以下である。本発明を実施した場合にはモード・レジスタ41内のB₁ に対応するモード・ビットは1、X₁ に対応するモード・ビットは0とすると、B₁ の指す汎用レジスタは64ビットの値、X₁ の指す汎用レジスタは下位32ビットに上位32ビットを0と見なした値が使用される。すなわち、モード・ビットが1の場合は対応する汎用レジスタの長さは64ビットとして扱い、モード・ビットが0の場合は対応する汎用レジスタの長さは32ビットとして扱われる。ここで、本実施例では、拡張モードでの長さを64ビットとしたが、任意のビット長さで

よいことを付記する。ただし、メモリとのアクセス単位がバイト(=8ビット)の場合には、8の倍数に設定するのがよい。またアドレス拡張をデータ拡張を別個のモード・ビットとして、各汎用レジスタに2ビットを対応させたモード・レジスタの構成としうることも付記する。ここでは、データの拡張の要請の説明で述べたように8バイトの固定小数点データの実現とあわせてアドレス拡張を行うため、64ビットとした。またデータ拡張としては64ビット・アドレス拡張としては(24ビット・アドレス方式のように)その一部(例えば下位48ビット)をアドレスとして使用する方式も同様にして実現できることを付記する。

第4図において、B₁、3、X₁、4、D₁、5の3者の和がメモリ上のデータにアクセスする64ビットのアドレス6として使用される。メモリからの読出しデータの長さはR₁ に対応するモード・ビットに依る。モード・ビットが1の時は、R₁ の指す汎用レジスタの64ビットに8バイトのデータ(Ⓐ、Ⓑの部分)が読出されてセツ

トされる。モード・ビットが0の時は、4バイトのデータ(④の部分)が読出され、④の最上位ビットxの値が0の時その上位に32ビットの0が付加された値が、またxが7の時1が付加された値が、それぞれ汎用レジスタ7にセットされる。

次に、本実施例の詳細な説明を行う。まず、汎用レジスタ制御部10の動作を第5図により説明する。汎用レジスタは16本あり(11)それらとの入出力はレジスタ番号指定レジスタ14により選択して実行される。すなわち、レジスタ番号指定レジスタ14の内容に従って16本の汎用レジスタ11の中の1つが選択され、書き込みの場合はレジスタ入力データ・レジスタ12を経由して、また読出しの場合はレジスタ出力データ・レジスタ13を経由して64ビット単位での書き込み/読出しが実施される。

次に、アドレス計算部20の動作を第6図を用いて説明する。第4図に述べたLOAD命令1の例について説明する。命令語によつて指定されたレジ

スタX₁の内容が第5図の汎用レジスタ制御部10から読出され、X₁レジスタ・データ・レジスタ21にセットされる。またX₁レジスタに対応するモード・ビット125がモード・レジスタ制御部40から読出される(詳細は後述)。X₁モード・ビット125の値が0の時には、0を保持している定数レジスタ22の内容がアドレス・アダー25のX₁。入力レジスタ26の前半32ビット・セットされる。X₁モード・ビット125の値が1の時には、X₁レジスタ・データ・レジスタ21の前半32ビットがX₁。入力レジスタ26の前半32ビットにセットされる。どちらの場合も後半32ビットにはX₁。レジスタ・データ・レジスタ21の後半32ビットの値がそのままセットされる。B₁についてもX₁について説明した上記の処理が同様にして行われ、B₁。入力レジスタ27にセットされる。命令語1のD₁。パートはそのままD₁。入力レジスタ28にセットされる。X₁。入力レジスタ26、B₁。入力レジスタ27、D₁。入力レジスタ28はアドレス・アダー

25の3入力となり、3者の加算が実施され、結果が出力レジスタ121にセットされる。選択器122は、X₁。モード・ビット125およびB₁。モード・ビット124がともに0の時には、アドレス・レジスタ123の前半32ビットには定数レジスタ22の内容をセットし、少くとも一方の1の場合には出力レジスタの前半32ビットの内容をセットするために用いられる。どちらの場合においてもアドレス・レジスタ123の後半32ビットは、出力レジスタ121の後半32ビットがそのままセットされる。以上の結果、X₁。レジスタ、B₁。レジスタの両者が従来モードの時には有効長32ビットのアドレスを生成し、少くとも一方が拡張モードの時には有効長64ビットのアドレスが生成される。生成されたアドレスに従い、メモリ・リクエストが実行されるが、その動作回路は公知なので省略する。

次に、演算制御部30の説明を第7図を用いて行う。先ず、LOAD命令の処理の概略を説明する。上述のメモリ・リクエストの結果として、メモリ

から読込まれたデータがメモリ・フェッチ・データ・レジスタ31にセットされる。その内容は、64ビットのALU32の入力レジスタ33にセットされるとともに、選択器34の入力となる。LOAD命令の場合には、選択器34によりメモリ・フェッチ・データがそのままモード選択器35の入力となる。メモリ・フェッチ・データ・レジスタ31の前半部はさらに、32ビットのALU36の入力レジスタ37の入力となるとともに、選択器38の入力となる。LOAD命令の場合は上と同様にメモリ・フェッチ・データがそのままデータ拡張器39の入力となり、さらにデータ拡張器39により64ビットのデータに拡張される。すなわち、32ビットのデータの最上位1ビットの値が1の時にはすべて1の32ビットのデータが付加され、0の時にはすべて0の32ビットのデータが付加される。このための回路も既知なのでここでは説明は省略する。データ拡張器39の出力は、前述のモード選択器35の入力となる。モード選択器35においては、命令語のR₁の指すレジ

タに対応するモード・ビットがモード・レジスタ制御部40より伝えられ、その値が1の時には、64ビット系のデータを、その値が0の時には32ビット系のデータを、それぞれ選択し、結果137をレジスタ入力データとして汎用レジスタ制御部10に転送する。汎用レジスタ制御部10では既述の如く汎用レジスタ11にセットする。以上のようにしてLOAD命令の処理(第4図で述べた処理)が完了される。次に、データが拡張の例としてADD命令の処理を本図を用いて説明する。本命令はLOAD命令1と同じRX形式であり、X、B、D、により指示されるメモリ上のデータをフェッチし、R、の指す汎用レジスタの内容と加算し、結果を元のR、の指す汎用レジスタにセットする命令である。メモリからフェッチされたデータはLOAD命令の時と同じ処理によりメモリ・フェッチ・データ・レジスタ31にセットされる。ADD命令の場合には、64ビットALU32と32ビットALU36の演算が実施される。両ALUの入力には、汎用レジスタ制御部10によ

り読出されたデータがレジスタ・データ・レジスタ131にセットされる。その全体64ビットは64ビットALUの入力レジスタ132に、下位32ビットは32ビットALUの入力レジスタ133にセットされる。ADD命令の場合には、LOAD命令のときとは異なり選択器34、38ではALU出力134、135が選択される。後者はデータ拡張器39により64ビットのデータに拡張される。選択器34の出力とデータ拡張器39の出力はモード選択器35の入力となり、R、モード・ビットの値に従ってどちらかが選択され、レジスタ入力データ137となる。この値137は汎用レジスタ制御部10において、R、の指す汎用レジスタにセットされる。このようにして、ADD命令が実行される。ここで、64ビットALU32は本発明で新たに設けたものであるが、32ビットALUを容易に拡張して実現できるので、詳細は省略するが、演算の長さが32ビットから64ビットになっただけであり、オーバフローなどの割込み要因発生回路も32ビットALU

と同様に先頭ビットからのオーバフローの検出などを行なえばできる。また、本図の64ビットALU、32ビットALU、データ拡張器などを一体とした同じ機能をもつALUを作成することも容易にできる。

次に、モード・レジスタ制御部の動作を第8図を用いて行う。命令語1のレジスタパート107がレジスタ番号44に設定され、セレクト42のセレクト信号となる。モード・レジスタ(MR)41はセレクト42の入力となり、指定されたレジスタ番号に対応する部分がモード・ビット110として出力され、上述したアドレス計算部20、演算処理部30に伝達される。

次にモード・レジスタ41の設定、変更、メモリへの転送のために設ける命令について第8図、第9図を用いて説明する。4つの命令を新たに設けるのが良い。4つの命令はすべて第1図の中のS形式である。S形式のアドレス指定部(B、D、)は、メモリ・アドレスを指したり、後述のSR演算、AND演算のデータとして用いられる。

(1) OR Mode Register (OMR) 命令

命令語で指定されたオペランドアドレスのうち下位16ビットの情報とモード・レジスタ41の情報の各ビット対応のOR演算が行われ(第9図のステップ201)結果がモード・レジスタに設定される(ステップ202)。すなわち、第8図において、デコーダ43によりNMR命令信号141がオフになる。ところで、MR41の内容はレジスタ142を経由してAND回路143、OR回路144の入力となる。また、オペランド・アドレスはレジスタ145を経由して両回路の入力となり、両者のAND演算、OR演算が行われ、結果はレジスタ145、146にそれぞれセットされる。上述のようにOMR命令の場合、NMR命令信号線はオフになっておりOR回路の出力146が選択される。さらにLMR命令信号線147はオフとなり、MR書き込み信号線148はオンとなるようにデコードされ、OR回路の出力146がMR41にセットされる。

(2) AND Mode Register (NMR) 命令

OMR命令において、OR演算がAND演算になる他は同じであり、NMR命令信号線141がオンとなるようにデコードされ、ANDの結果145がMR41にセットされる。

(3) Load Mode Register (LMR) 命令

命令語で指定したアドレスのデータ2バイトをメモリより読込み(ステップ205)、モード・レジスタ41に設定する(ステップ206)。すなわち、メモリ・フェッチ・データ・レジスタ149の上位16ビットをMR41に書き込む。デコーダ43において、LMR命令信号線147がオンになり、メモリ・フェッチ・データ・レジスタ149の入力を選択し、MR書き込み信号線148をオンになることによりMR41に書き込まれる。

(4) Store Mode Register (SMR) 命令

モード・レジスタ41の内容が、本命令語で示されたアドレスにストアされる。本命令の場合、デコーダ43によりSMR命令信号線150

がオンになり、メモリ・ストア・データ・レジスタ151を経由して、メモリ・リクエストが実行される。

以上、モード・レジスタ41に関する4つの命令の動作を説明したが、これらの命令は非特権命令とした方がよい。すなわち、汎用レジスタの内容は、プログラム毎に変化し、プログラム内においても時々刻々変化するもので、非特権プログラムにおいても上記の命令が使用可能とするためである。また、モード・レジスタの初期値はすべて0とするのがよい。

以上、本実施例の説明を行った。本実施例によれば、次のような効果がある。

(1) 従来のプログラムを変更しなくても、そのまま実行することができる。モード・レジスタの初期値は0なので、従来の有効長として扱われるので、従来のプログラムの変更はまったく不要である。

(2) 従来のプログラムを変更して、データ・アドレスのみ拡張アドレス部分にアクセスするように

するには、拡張アドレス部分にアクセスする前に、拡張アドレス用のレジスタとして使われる汎用レジスタに対応するモード・ビットを1にして、メモリ・アクセスを行い、拡張アドレス用のレジスタとしての使用が終了した時点でモード・ビットを0にするような変更、すなわち2命令の追加を行えばよいので容易にアドレス拡張が実現できる。

(3) 従来のプログラムを、拡張アドレス部分に格納して動作させる場合においても、プログラム部分のベース・アドレスとして使われる汎用レジスタに対して、上記と同じような変更を加えるだけで済む。

(4) 固定小数点データの処理単位を従来の32ビットから64ビットに拡張することも上記と同様に拡張データ用に使用する前に汎用レジスタのモード・ビットを1にするだけで、従来の同じ命令体系の命令語を使用して、64ビットの固定小数点の演算ができる。

(5) 新規に拡張アドレスで動作するプログラムや

拡張データを扱うことができるプログラムを作成する際においても、従来のコンパイラなどのソフトウェア技術をほとんどそのまま活用することができる。

なお、拡張モードで使用されている汎用レジスタの内容をメモリに記憶したり、メモリから値を設定する場合には、そのための格納領域の大きさを従来の4バイトから8バイトに変更する必要があるが、その操作は容易である。

〔発明の効果〕

本発明によれば、命令体系の変更は殆ど行わずに、アドレス拡張やデータ拡張が実現できるので、従来のプログラムの流用、および従来のコンパイラなどのソフトウェアの流用が容易になるという効果がある。すなわち、(1)命令体系が殆ど変わらず、(2)汎用レジスタが拡張アドレスまたは拡張データとして使用される時のみ、対応するモード・ビットの変更を行う命令を追加すれば、アドレス拡張ならびにデータ拡張が実現できるので、過去に蓄積された膨大なプログラムをそのまま実行

することはもちろん、若干の修正のみで拡張アドレスで動作させたり、拡張データを扱ったりすることができるようになる。以上の結果、アドレス拡張やデータ拡張に伴ない必要となるソフトウェアの開発工数を非常に小さくできるという効果がある。

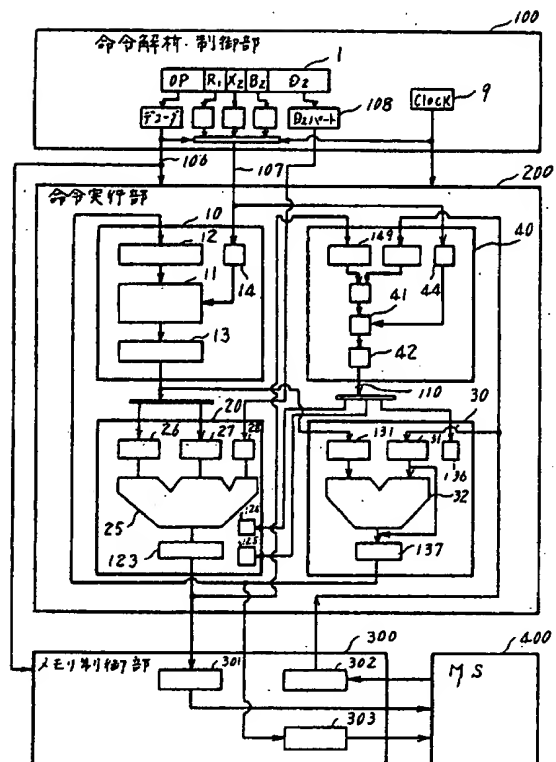
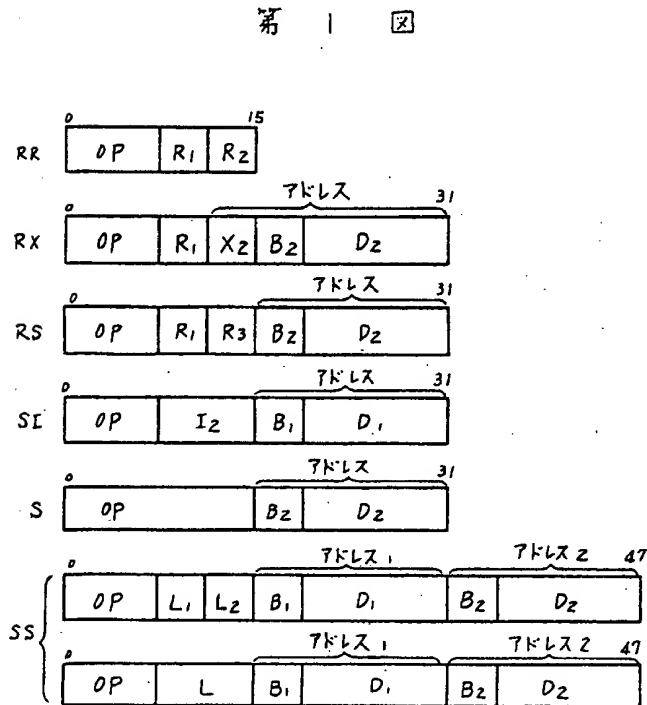
図面の簡単な説明

第1図は情報処理装置の命令形式の説明図、第2図は本発明の実施例の構成図、第3図はLOAD命令、ADD命令の動作概要図、第4図はLOAD命令の詳細な動作図、第5図は汎用レジスタ制御部の回路構成図、第6図はアドレス計算部の回路構成図、第7図は演算処理部の回路構成図、第8図はモード・レジスタ制御部の回路構成図、第9図はモード・レジスタ演算命令の動作の説明図である。
10…汎用レジスタ制御部、11…汎用レジスタ、20…アドレス計算部、25…アドレス・アダー、30…演算処理部、32、36…ALU、40…モード・レジスタ制御部、41…モード・レジスタ、23、24、134、38…セクタ、35

…モード選択器、39…データ拡張器、143…AND回路、144…OR回路、100…命令解析制御部、200…命令実行部、300…メモリ制御部、400…主記憶装置。

代理人 弁理士 高橋明夫

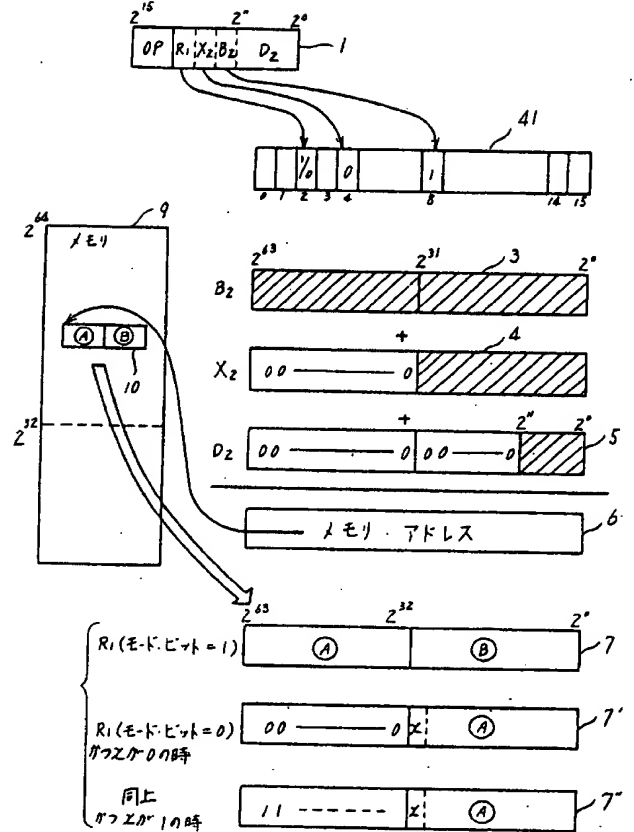
第 2 図



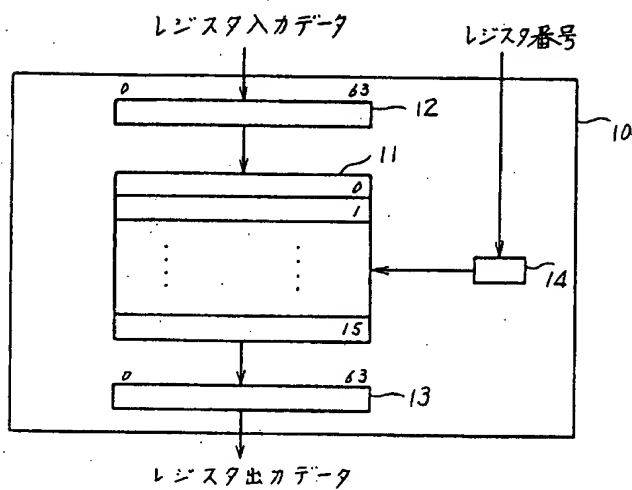
第 3 図

ステージ名	LOAD 命令	ADD 命令
D 命令デコードステージ	命令語内のオペレーション・コードによりLOAD命令と認識する。	命令語内のオペレーション・コードによりADD命令と認識する。
A アドレス計算ステージ	命令語内の B_2, X_2, D_2 パートを基にオ2オペランドのアドレスを計算する。 ① B_2 モードビットと B_2 レジスタ・データを讀出す。 ② X_2 モードビットと X_2 レジスタ・データを讀出す。 ③ B_2 レジスタ・データ, X_2 レジスタ・データ, D_2 パートの3者の加算結果をアドレスとする。	
L オペランドロードステージ	オ2オペランドのデータをメモリからフェッチする。 ① Aステージの結果のアドレスを用いて、メモリ・フェッチを行なう。 ② フェッチしたデータをメモリ・フェッチ・データにセットする。	
E 命令実行ステージ	命令語内の R_1 パートの指定レジスタに、Lステージでフェッチしたデータをセットする。 ① R_1 モードビットを讀出す。 ② R_1 モードビットの値に従って、フェッチしたデータをセットする長さを変更する。	命令語内の R_1 パートの指定レジスタの内容と、Lステージでフェッチしたデータの加算を行い、結果を当該レジスタにセットする。 ① 同左 ② 同左

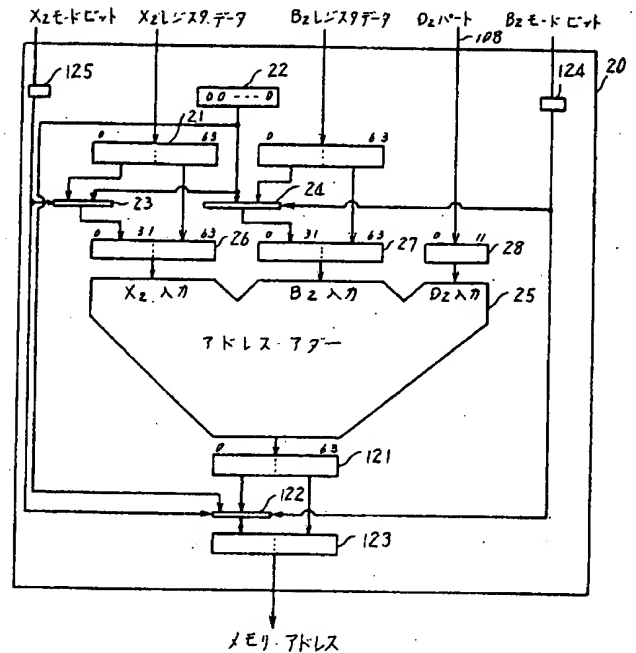
第 4 図



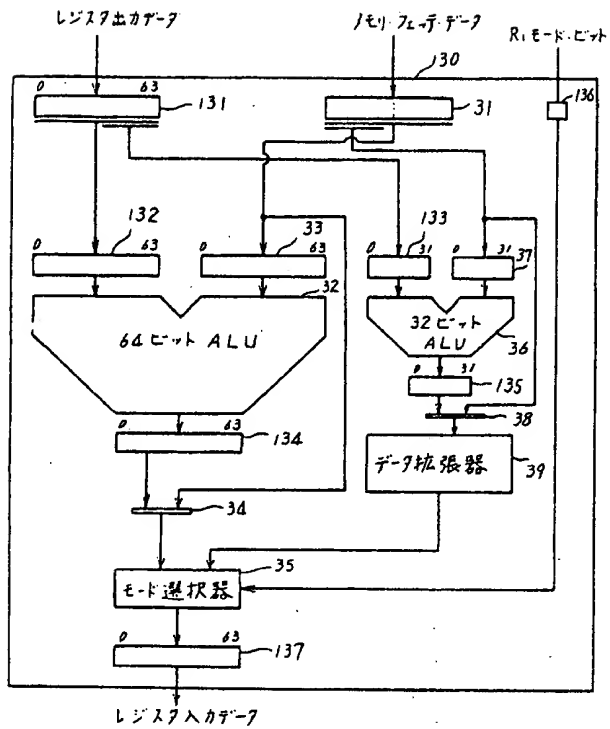
第 5 図



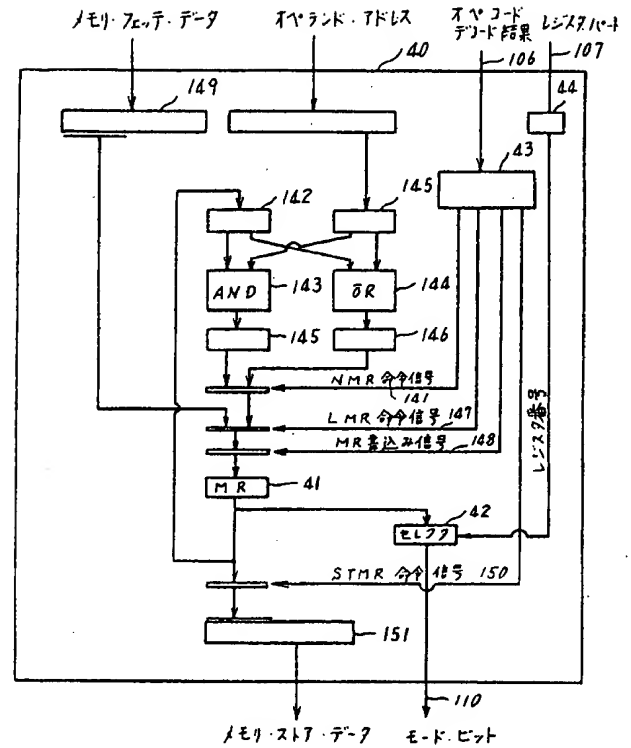
第 6 図



第 7 回



第 8 题



第 9 回

